

Fig. 1

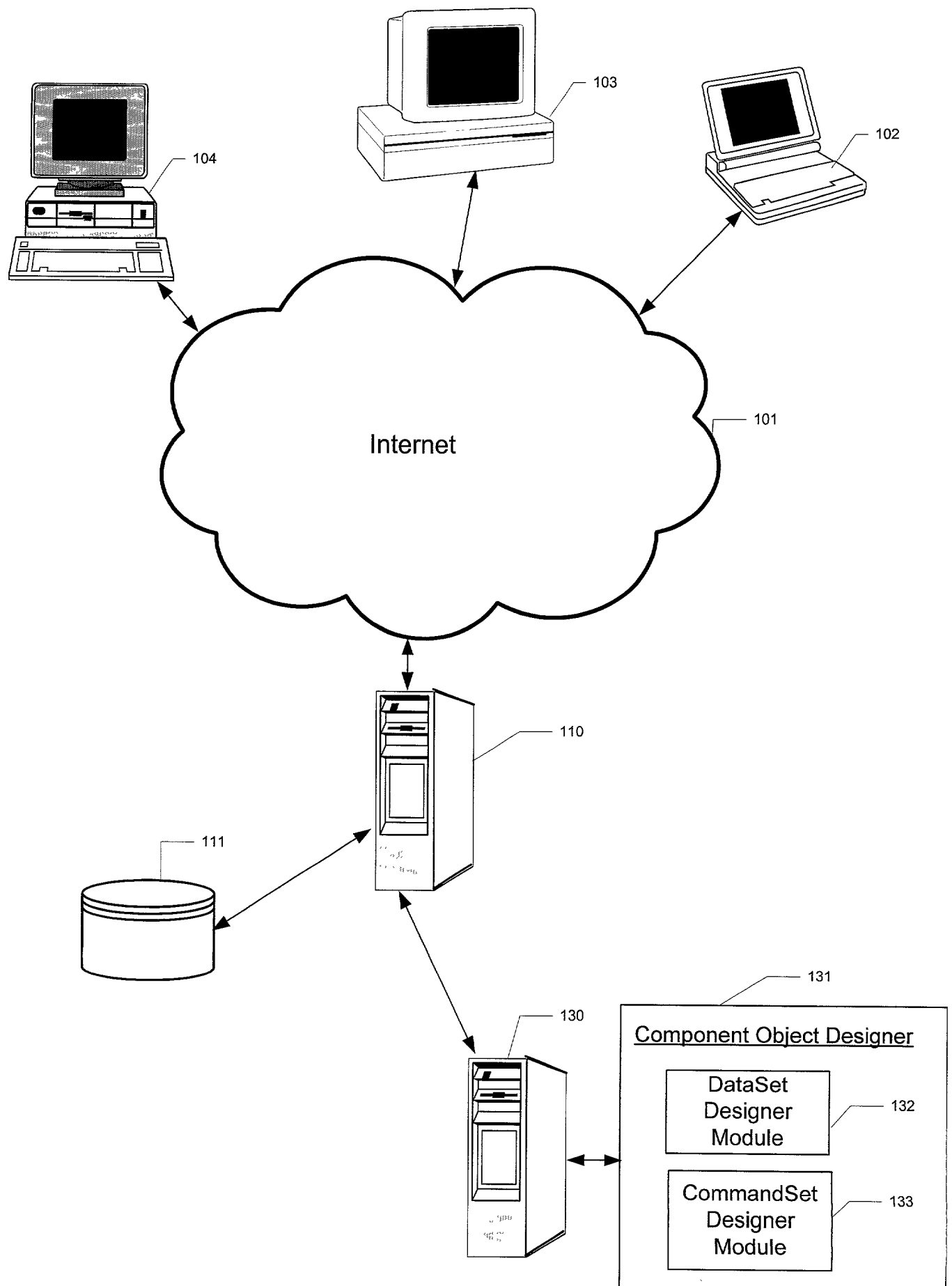
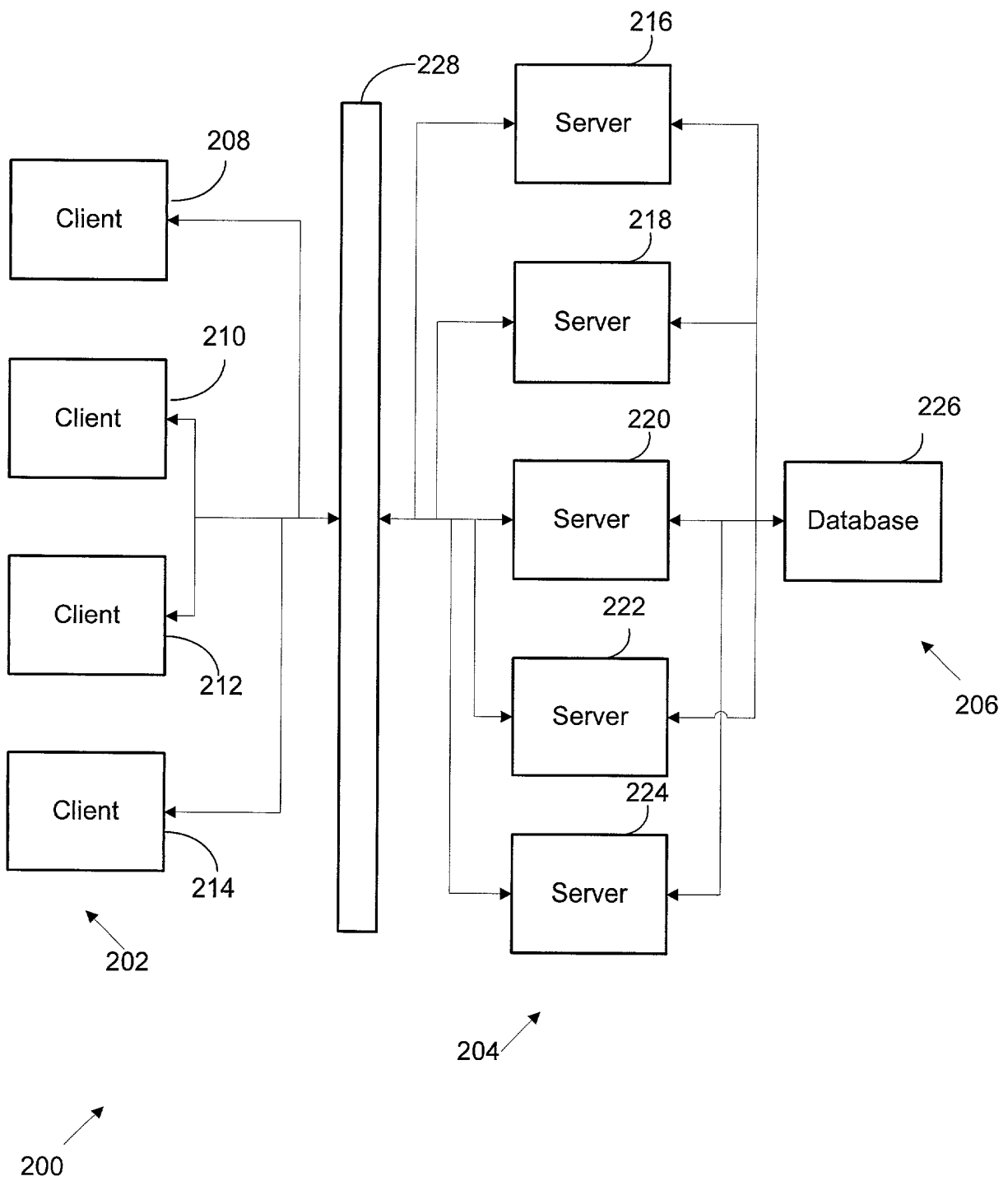


Fig. 2



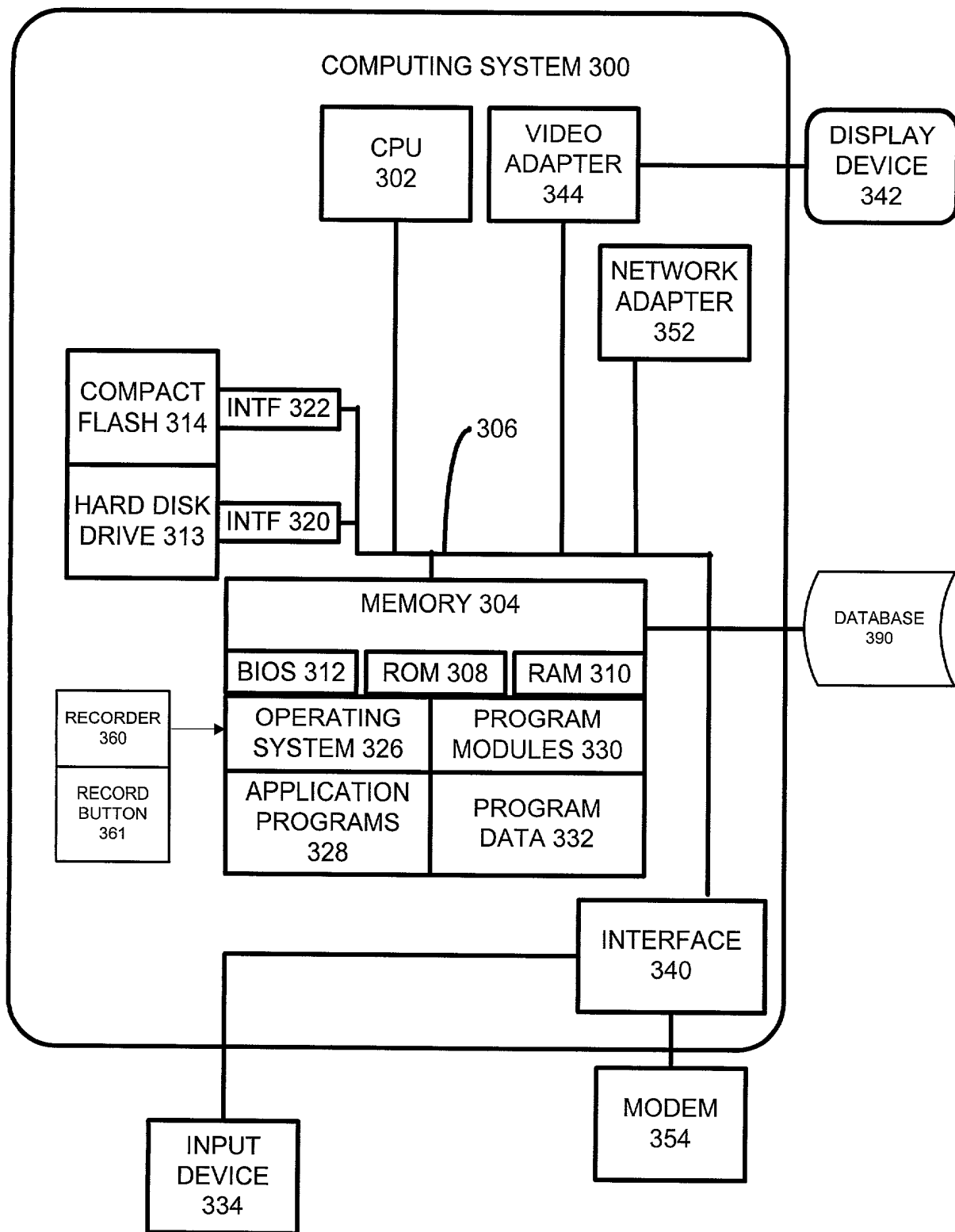


Fig. 3

Fig. 4

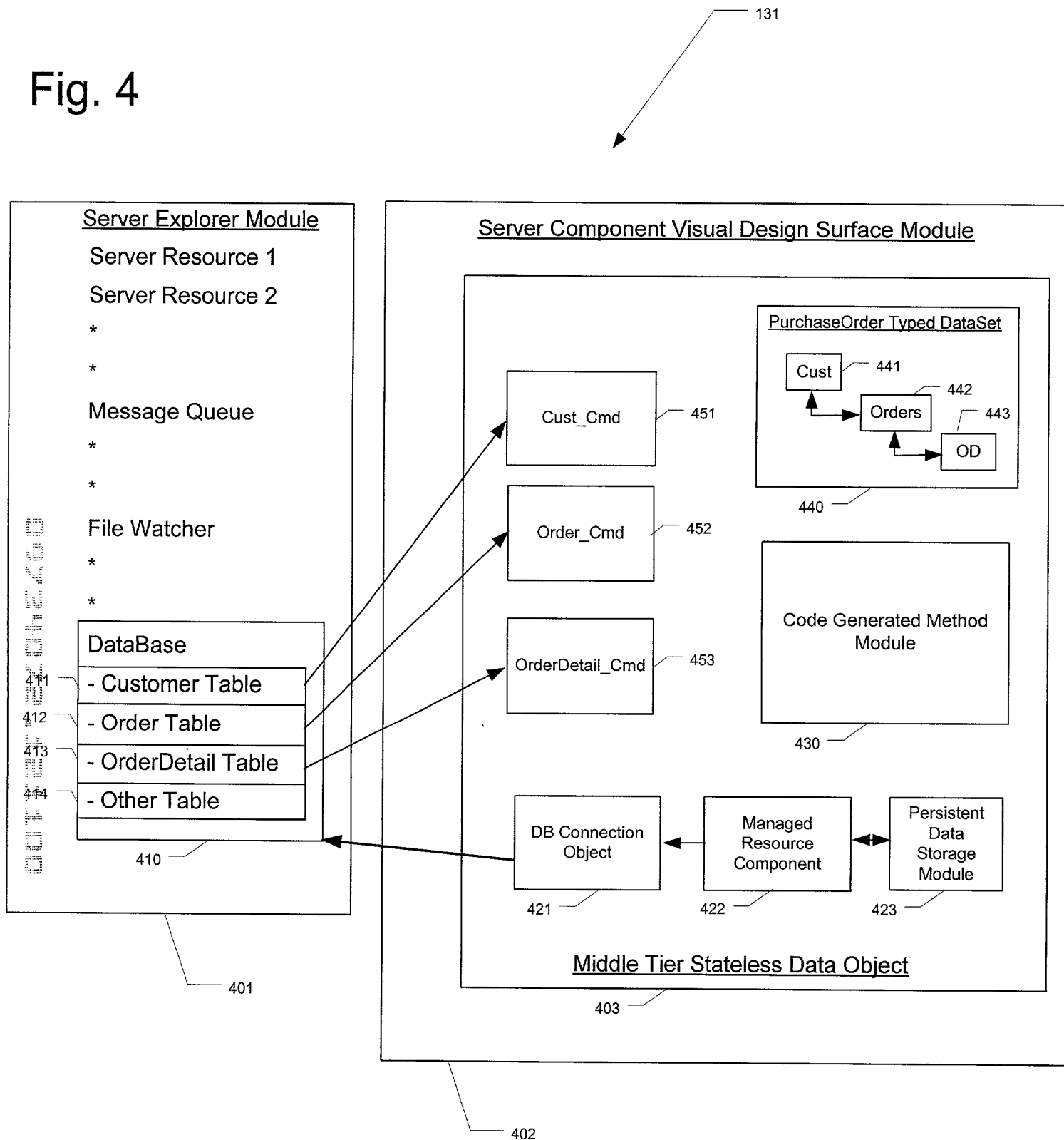


Fig. 5

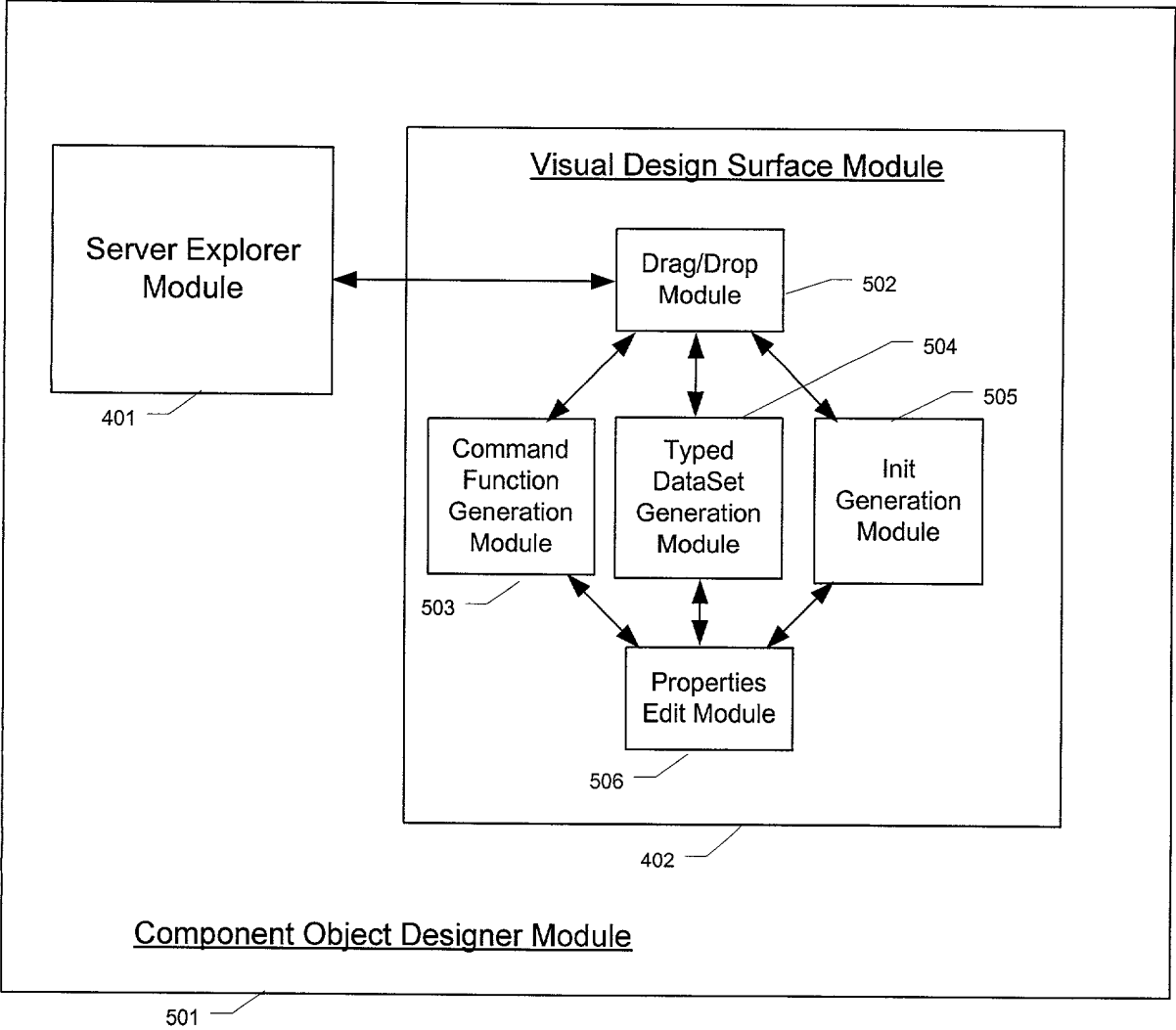


Fig. 6

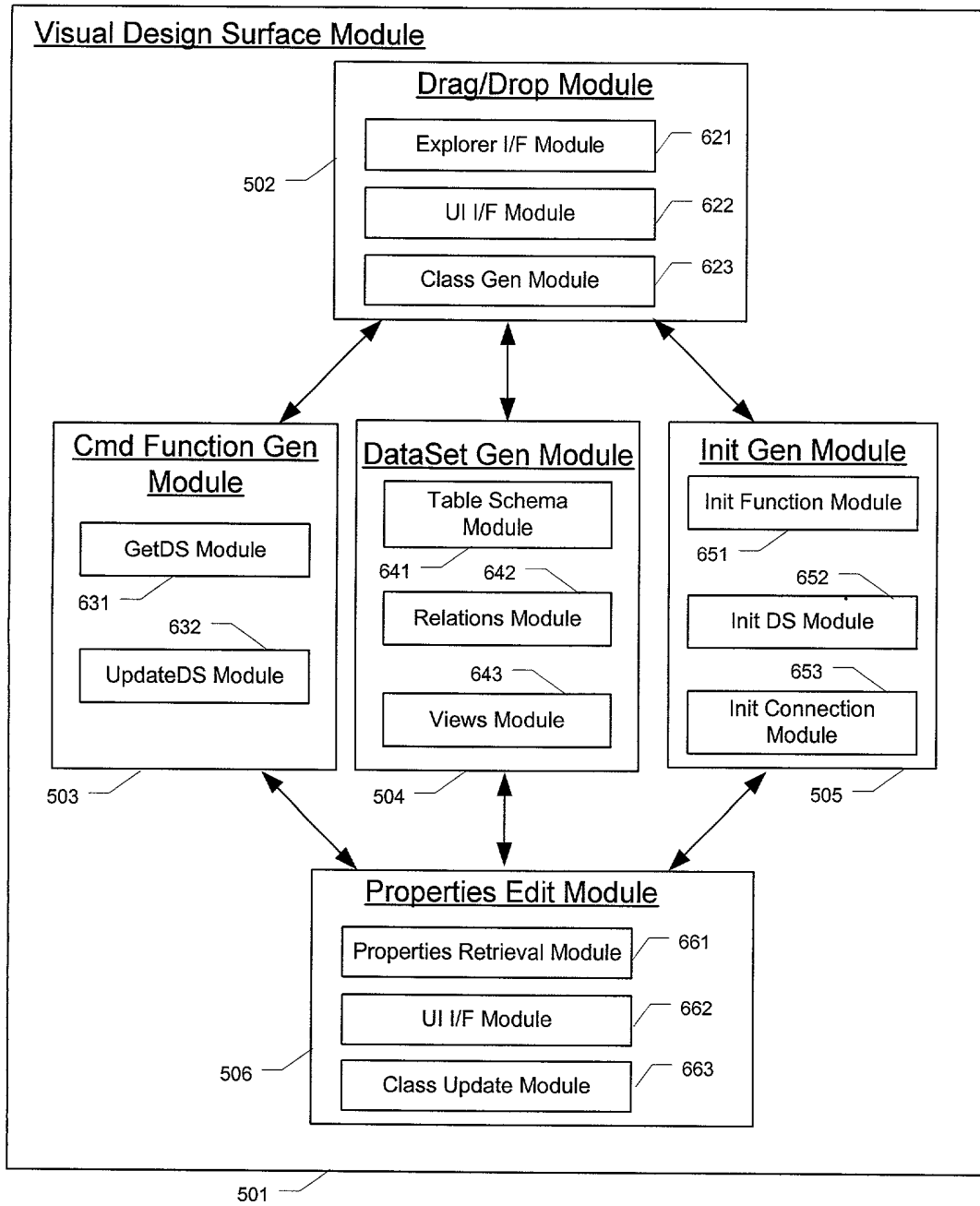


Fig. 7a

FillDataSet Method

Public Function XXXXFillDataSet(ByVal dataSet As DataSet, ByVal {ParameterName} As System.{ParameterType})) As Long

' Copy input Parameter Values to the Parameter Collection

DIM vtParameter **As Variant**

vtParameter1 = {ParameterName}

Set {AdapterInstance}.SelectCommand.Parameters.Item(ParameterName.Value=vtParameter1

'Create an error handling block in case filldataset throws an exception

Try

'Load dataSet from the command

XXXXXFillDataSet = {AdapterInstance}.**FillDataSet**(dataset)

Catch eFillException as Exception

'if we can do a debug.print without a reference, then we should print the

'error info to the output window and re-throw the error.

End

701

End Function

Fig. 7b

UpdateDataSet Method

Public Function XXXXUodateDataSet (ByVal updateDataSet As DataSet) As Long

const DataCursorRowState flag = DataCursorRowState.new | DataCursorRowState.Deleted |

DataCursorRowState.ModifiedCurrent;

DataRow[] datarows = dataTable.Select(" ", " ", flag);

Dim UpdateRows as DataRows()

Dim NewRows as DataRows()

Dim DeletedRows as DataRows()

' Get All of the Deleted Rows and Update the Data Store

DeletedRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.Deleted)

If 0 < dataRowLength then

Update(DeletedRows, AdapterInstance.DefaultSourceTableName);

End If

' Get All of the Updates Rows and Update the Data Store

UpdatedRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.ModifiedCurrent)

If 0 < dataRowLength then

Update(DeletedRows, AdapterInstance.DefaultSourceTableName);

End If

' Get All of the New Rows and Update the Data Store

NewRows = UpdateDataSet.Select(" ", " ", DataCursorRowState.New)

If 0 < dataRowLength then

Update(DeletedRows, AdapterInstance.DefaultSourceTableName);

End If

End Function

702

Fig. 8a

FillAllDataSet

```
Public Sub FillAllDataSet(ByVal dataset As DataSet(, ByVal Parameters As System.{ParameterType})
```

```
    ' The line below is added for each adapter on the Virtual Design Surface
```

```
    ' Passing into the function only the necessary parameters
```

```
    (AdapterName1)FillDataSet dataset{, _parameters}
```

```
    (AdapterName2)FillDataSet dataset{, _parameters}
```

```
    *
```

```
    *
```

```
    *
```

```
    (AdapterNameN)FillDataSet dataset{, _parameters}
```

```
End Sub
```

802

801

Fig. 8b

UpdateAllDataSet

```
Public Sub Update{DataSetName} (ByVal dataset As DataSet)
```

```
    ' The line below is added for each adapter on the Virtual Design Surface
```

```
    (AdapterName1)Update dataSet
```

```
    (AdapterName2)Update dataSet
```

```
    *
```

```
    *
```

```
    *
```

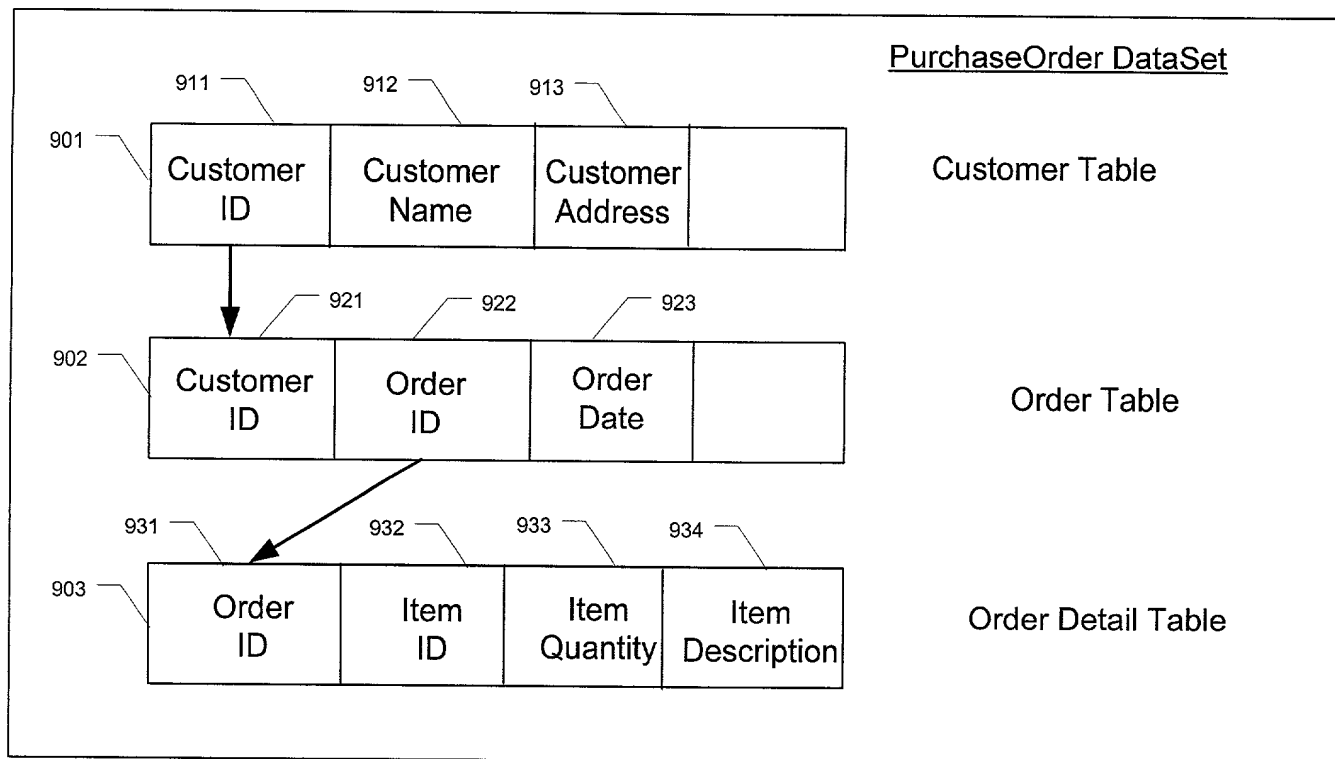
```
    (AdapterNameN)Update dataSet
```

```
End Sub
```

812

811

Fig. 9



Public Sub FillAllDataSet(ByVal PurchaseOrder As DataSet(, ByVal Parameters As System.{ParameterType}))

CustomerFillDataSet PurchaseOrder(, _parameters}
 OrderFillDataSet PurchaseOrder(, _parameters}
 OrderDetailFillDataSet PurchaseOrder(, _parameters}

End Sub

Public Sub UpdateAll (ByVal PurchaseOrder As DataSet)

' The line below is added for each adapter on the Virtual Design Surface

OrderDetailUpdate (PurchaseOrder, Filtered (Delete))
 OrderUpdate (PurchaseOrder, Filtered (Delete))
 CustomerUpdate (PurchaseOrder, Filtered (Delete))

' The line below is added for each adapter on the Virtual Design Surface

OrderDetailUpdate (PurchaseOrder, Filtered (Insert | Update))
 OrderUpdate (PurchaseOrder, Filtered (Insert | Update))
 CustomerUpdate (PurchaseOrder, Filtered (Insert | Update))

End Sub

Fig. 10

